Electronic Notes in Theoretical Computer Science 23 No. 1 (1999)
URL: http://www.elsevier.nl/locate/entcs/volume23.html 2 pages

Process Realizability (Abstract)

Samson Abramsky

Laboratory for Foundations of Computer Science, The University of Edinburgh, James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, Scotland, U.K.

samson@dcs.ed.ac.uk http://www.dcs.ed.ac.uk/home/samson/

Abstract

We introduce a notion of realizability for Classical Linear Logic, and describe a number of examples, including one based on concurrent games, and one based on the process calculus CCS.

Keywords: Realizability, Linear Logic, Process Calculi, Concurrency, CCS, Concurrent Games, Semantics.

MSC: 03B40, 03D65, 68Q55.

Realizability in its usual formulations, while useful and powerful, appears limited in its scope. In particular, it seems closely tied to Intuitionistic logic, on the logical side, and to functional computation, on the computational side. We propose a significant broadening of the scope of realizability: on the logical side, to Classical Linear Logic [Gir87], and on the computational side to encompass concurrent processes. Examples include a realizability model for Classical Linear Logic based on the process calculus CCS [Mil89]. The guiding principle for this notion of process realizability is that of *duality*: a type is understood not just in terms of its canonical "values", but also in terms of the contexts for consuming these values; a formula is understood, not just in terms of its possible proofs, but also of its possible refutations; an interactive protocol is understood in terms of both "clients" and 'servers"; etc. The realizability models for Classical Linear Logic are *not* well-pointed—an essential feature from our point of view. They also offer an interesting and novel basis for rejecting the structural rules of contraction and weakening—so that Linear Logic itself can be seen as arising naturally from this realizability paradigm.

One example of this construction, based on "concurrent games", has led to full completeness results for Multiplicative-Additive Linear Logic [AM99]. The example based on CCS suggests a number of possible applications to concurrency. It can be seen as a continuation of the work on interaction categories [AGN96a,AGN96b], which overcomes some of the problems which were

encountered there. There are also possible applications to the computational interpretation of proofs in Classical Logic.

References

[AGN96a] S. Abramsky, S. Gay and R. Nagarajan. Interaction Categories and the Foundations of Typed Concurrent Programming. In *Proceedings of the 1994 Marktoberdorf Summer School on Deductive Program Design*. M. Broy, ed. Springer-Verlag 1996, 35–113.

[AGN96b] S. Abramsky, S. Gay and R. Nagarajan. Specification Structures and Propositions-as-Types for Concurrency. In *Logics for Concurrency: Structure vs. Automata—Proceedings of the VIIIth Banff Higher-Order Workshop*. G. Birtwistle and F. Moller, eds. Springer-Verlag 1996, 5–40.

[AM99] S. Abramsky and P.-A. Melliès. Concurrent Games and Full Completeness. In *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, 1999.

[Gir87] J.-Y. Girard. Linear Logic. *Theoretical Computer Science* 50(1) 1–102, 1987.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice Hall International, 1989.